2.9 DECISION-MAKING CAPABILITIES

Decision-making elements (DMEs) simulated within a mission-level combat model implement updating, assessing, and controlling the tactical situation. Updates are done through data fusion and correlation; assessments result from identifying and prioritizing targets; control is exercised through avoiding, engaging, and delegating targets. The modeled decision-making elements represent either human decision-makers such as pilots, radar-operators, or a ship's commanding officer, or they represent automated decision-makers such as autopilots or internal missile guidance systems.

Simulated decision-making elements will be discussed in three aspects: capabilities, knowledge base, and logic processes. Capabilities modeling is discussed in this section and includes the decision-maker's thinking functions or activities, and they may include a servicing time for the decision. The knowledge base is discussed in Section 2.10 and includes both tactical and situational knowledge. Tactical knowledge includes scenario-dependent tactical objectives, preplanned decisions, and the information available to the decision-maker at the start of the exercise; situational knowledge is acquired by the decision-maker as the exercise unfolds. Logic processes are discussed in Section 2.11 and describe how the decision-maker develops and maintains his perceptions of the tactical situation, assesses threats to the tactical situation, makes decisions based on the assessment, and selects actions.

The decision-making capabilities within a mission-level combat model include the decision-making systems, the decision-making activities each decision-maker is capable of performing or authorized to perform, and each decision's servicing time for the specific decision-maker. The parent player's responsibilities in the scenario determine the thinking activities assigned to each decision-maker as well as the servicing times for each designated thinking activity. Each decision-maker within a scenario will probably not engage in every decision-making activity. For example, a submarine diving officer does not have the authority to issue the torpedo fire order, but he will have thinking activities appropriate to changing the submarine's depth and rate of descent/ascent. Also, two decision-makers engaged in the same activity may have different servicing times; the commanding officer may require 0.9 seconds to issue the torpedo fire order, while the weapons officer may require 1.25 seconds.

2.9.1 Functional Element Design Requirements

The design requirements for the decision-making capabilities functional element are:

a. Provide the user with the capability to define decision-making or thinker systems within the player-structure.

b. Provide the user with the capability to assign appropriate thinking activities to each thinker system.

c. Provide the user with the capability to define the servicing or thinking times for each designated thinking activity for a specific thinker system.

2.9.2 Functional Element Design Approach

The thinker is the simplest system to describe in Suppressor input. A thinker system is required in a PLAYER-STRUCTURE for that player to be able to make decisions and perform actions (referred to as TACTICs in Suppressor). Also, a thinker system is required to recognize and assimilate data received from sensors.

In Suppressor, a thinker system is modeled as a group of delay times. These delay times are associated with thinking processes, most of which are described in detail in the functional elements which follow. The list of possible entries (found in the thinker system's TIME-TO-THINK data item) is below:

TABLE 2.9-1. Suppressor-Defined Thinking Activities.

Item	Defines how long it takes the thinker to:
ASSIMILATE-INTELL	assimilate sensor or message information into the target perception
CONSIDER-ASG/CANCEL	decide to make or cancel a target assignment
CONSIDER-LAUNCH	decide whether to issue a launch order
CONSIDER-MOVE	decide whether to make a reactive movement
EVAL-ASSIGN-THREAT	decide to add or drop a subordinate from a target's assignment queue
EVAL-EMCON-CHANGE	turn sensors on/off via emission control
EVAL-ENGAGE-THREAT	decide to add or drop a weapon from a target's engagement queue
EVAL-FIRING	decide to start or stop a firing sequence against a target
EVAL-JMR-QUEUE	decide to add or drop a jammer from a target's jammer queue
EVAL-JMR-SPOTS	decide to add or drop jammer spots focused against a target
EVAL-LETHAL-ENGAGE	decide to start or stop an engagement against a target
RECOG-MSG	recognize receipt (or non-receipt) of a message from another player
RECOG-PHYS-EVENT	recognize an attack on the target
RECOG-SNR-EVENT	recognize results of a sensor chance
REVIEW-INFORMATION	review all current target perceptions to see if they need to be dropped

Suppressor usually simulates a thinking process as two separately scheduled events. The first is a dummy event whose only purpose is to schedule the second event. The time lag between the two events is equal to the value of the TIME-TO-THINK entry corresponding to the thinking process.

In the default case, a thinker is able to handle one thinking process at a time. When the scenario becomes intense, this could cause the thinker system to get backed up, causing thinking events to not be processed in a timely manner. Two remedies are available to the scenario developer for this situation. He may place multiple thinker systems on the player or he may use the MAX-CONCURRENT-EVENTS thinker system data item. Specifying a value greater than one will multiply the throughput of the thinker system by that factor, thus allowing several events to be processed simultaneously by that thinker.

It should be noted that if Suppressor tries to initiate a thinking process for which a TIME-TO-THINK entry is not present, then a non-fatal warning will be presented to the user indicating that the thinking process could not be performed.

2.9.3 Functional Element Software Design

The executive routine for thinking events is SIMTHK. Its design is:

```
ABSTRACT     executive control over mental events                    |
    *begin logic to process mental events:
       *when external input data:
          *invoke logic to add entry to pending queue;
       *otherwise, input is self-scheduled:
          *set thinking element status to not busy;
          *recycle decision buffer;
          *when perceived data decisions (331xxx):
             *invoke logic to add digest entry to pending queue;
          *but, when event is information fusion (332[1,2]xx):
             *invoke logic to next action with digested info;
          *but, when command-related info (3323xx):
             *invoke logic for command-related fusion;
          *but, when resource allocation (333xxx):
             *invoke logic for general purpose reactions;
          *but, when information review (334100):
             *invoke logic for reviewing perceptions;
          *but, when movement contingency plans (335100):
             *invoke logic to ponder movement options;
          *end of test for type of data input.
       *end of test for mental event.
       *loop, until all thinking capability properly tasked:
          *invoke logic to determine what should be done now;
          *when anything to process or think about:
             *when recognize communications (321100):
                *invoke logic to perceive communications input;
             *but, when explicit sensor input (3212xx):
                *invoke logic to perceive explicit sensor input;
             *but, when implicit physical stimuli (321300):
                *set output to input, reset event code;
             *but, when data assimilation (322xxx):
                *look up digested data header and related information;
                *when direct or indirect source intell:
                   *allocate digest intell header block;
                   *search for target on perception list;
                   *when direct source intelligence:
                      *store IFF results in perception block;
                      *reset results if target not sensed;
                   *end of test for direct source of intelligence.
                *end of test for direct or indirect source.
                *recycle digest data header;
             *but, when resource allocation (323xxx):
                *invoke logic to set up allocation lists;
             *but, when information review or ponder (32[4,5]100):
                *set pointer and event code;
             *end of test for type of data to be processed.
             *when an event should be scheduled:
                *possibly set thinking element to busy status;
                *allocate decision buffer;
                *invoke logic to add event to action item list;
             *end of test for event test.
             *recycle queue entry block;
          *end of test for check for something to process.
       *end of loop for thinking capability allocation loop.
    *end of logic for SIMTHK.
```

2.9.4 Assumptions and Limitations

None.

2.9.5 Known Problems or Anomalies

None.